

必读： 六大自动驾驶车辆使用案例

必读：六大自动驾驶车辆使用案例

汽车解决方案依赖电子系统在车辆中部署自动驾驶功能。通过这种功能，电子系统可以满足实时处理要求，并与多种高带宽和低带宽传感器相连接。自动驾驶功能还可实现防盗和安全功能部署，阻止未经授权进入和改装，并确保安全操作和柔性降级。

在自动驾驶车辆空间内，必须开发几个关键的系统：

- 自动驾驶视觉系统
- 车用雷达
- 传感器聚合和处理
- 安全认证
- 乘员监测
- 车辆间和车辆与基础设施间通信

本电子手册探索了工程师在建立这些关键系统时面临的挑战，并确定 Xilinx 技术和生态系统如何帮助他们克服此类挑战。

目录

汽车应用中的视觉系统	5
ADAS 雷达	9
汽车应用中的聚合、融合和加速	14
汽车功能安全	19
乘员监测	24
车联网 (V2X) 通信.....	28

有关安富利的更多信息，请访问
www.avnet.com/apac



简介

汽车从18 到 19 世纪问世以来，创新率和采用率的增长速度都越来越快，汽车安全和技术呈光速般进步。

今天的自动驾驶车辆是个十分复杂的系统，按不同的自主等级工作，车厢内外都有集成AI。四周的摄像头和传感器同时生成并处理大量的数据。然而，复杂性带来了很大的挑战；这些系统必须具备实时性能、可靠性并能够相互通信。

Xilinx 汽车级（XA）器件能够满足当前和未来自动驾驶车辆所要求的对接、性能、安全性、延迟和决断力方面的挑战。各种单片系统，例如本电子手册中介绍的 XA Zynq UltraScale+ MPSoC，在单器件中提供了处理系统和可编程逻辑，并具备任意互连功能以及使机器学习推理之类功能加速的能力。

单片系统内部



图像：Xilinx Zynq UltraScale+ MPSoC 架构

Zynq UltraScale+ MPSoC 内部的处理系统包含：

- 应用处理单元（APU）
- 实时处理单元（RPU）
- 平台管理单元（PMU）
- 配置安全单元（CSU）

16nm 可编程 MPSoC 架构包含可配置逻辑块、块随机存储器、运行频率可达 890MHz 的 DSP 元件以及其他单元。

任意互连

Xilinx SoCs 为开发人员提供各种可从处理系统存取的对接解决方案。其中包括用于通信和传感器的低速、低带宽接口（例如，SPI、I2C 和 UART）乃至高带宽、高速接口（例如，USB、SATA 和 PCIe）。正确的 PHY 可编程逻辑可提供任意互连功能，以便于轻松地部署定制、传统和快速发展的标准。

减少 SWaP-C

处理系统与可编程逻辑之间的紧密契合让设计人员能够创建符合汽车系统中尺寸、重量、功率和成本（SWaP-C）方面苛刻要求的解决方案。此类集成可缩小电路板面积并减少对多种器件的需求，同时通过减少芯片间的传输和有源器件的部署，降低功耗。

帮助汽车设计人员作出改变

Xilinx 提供广泛的可编程技术与强大的工具、IP 库和框架的生态系统的组合，旨在帮助设计人员适应不断演进的自动驾驶车辆蓝图。

Xilinx 汽车级器件根据 AEC-Q100 进行了部件认证筛选。这种部件品质的提升能够提高整体系统的可靠性并支持安全功能。



汽车应用中的视觉系统

简介

自动驾驶车辆功能需要车辆了解其所处的环境。从本质上了解其所处环境的最常见方法之一是使用嵌入式视觉系统。

这些视觉系统让车辆的系统不仅能够了解其周围的情况，还能处理图像以检测车辆和物体并将其归类。这种能力是许多自动驾驶操作的基石，包括交通堵塞辅助、自动泊车以及停止/开启高速公路驾驶和更高的 SAE 等级。

架构功能和考虑因素

常见的车辆架构是使用连接到摄像头处理模块的多个摄像头。该摄像头处理模块将会先完成摄像头控制、图像捕获和预处理，然后再实施更高等级的算法，例如机器学习推理。

然后，与适当的车辆系统交流这些算法的输出以采取相应的操作。通常，从图像中提取的数据将传送到车辆的中央处理系统进一步处理，并融合这些数据与其他传感器的数据，然后再做出影响车辆控制的决定，例如行驶或刹车。

安全性与可靠性

鉴于使用汽车视觉系统的应用性质，必须考虑安全性和可靠性。这通常是指符合 ISO 26262 安全标准，因此系统的架构需要优先考虑安全性。

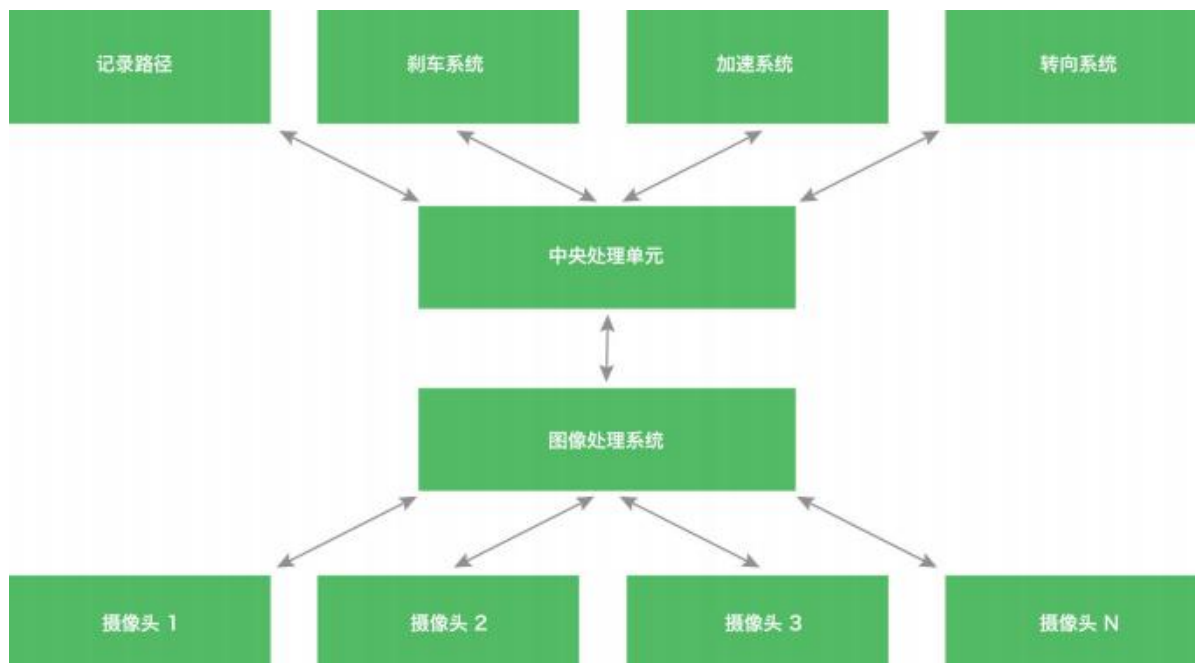


图 1: 普通车辆中的视觉系统架构

在车辆应用中，尽可能减少布线是一大关键，因为它会显著增加重量和成本。摄像头因其有宽输出总线、控制通道和电源走线，尤其是在需要部署多个摄像头时，要减少布线就很困难，因此制造商往往使用基于 MIPI CSI-2 标准的摄像头。这种方法可以减少布线，同时仍能获得高分辨率、高帧频摄像头所需的高数据率。为了实现此目的，制造商利用千兆位多媒体串行链路（GMSL）驱动器和接收器，为每个摄像头和电源、控制和视频提供一条单同轴电缆。这些使用中的驱动器和接收器在转换 GMSL 与 MIPI 之间，能实现与各种摄像头和处理系统的对接。

此类功能要求能够对接并控制多个摄像头，同时还能够控制 GMSL 链路。当然，系统还必须能够使用 CAN、CAN-FD、汽车以太网和 FlexRay 之类的标准汽车接口与下游车辆控制系统通信。

如果能够对接多种高速 MIPI 接口，同时提供必需的下游通信接口，则可在设计和器件中获得要求的对接能力。通常，这种折衷方法在于利用现成解决方案对接并达到对应的性能，还有通过创建自定义解决方案以不菲的支出获得要求的对接和性能。实施 MIPI DPHY 相当困难，因为它需要在同一个输入/输出单元中支持低摆幅高速差模信号和单侧低速控制信号。

处理并从传感器提取信息的功能需要出色的处理能力，以使车辆能够及时地做出决定。因此需要图像处理系统提供低延迟、高决断力的回应。通常，会因此而需要专用的高性能处理器或 ASIC 来达到这些要求。架构上的决定，例如使用外部 DDR 内存来存储处理管道中的图像段落，将会显著影响总体延迟和决断力，原因在于 DDR 作为共享系统资源的性质以及存取所需的时间。

在处理器上运行的是软件框架，它执行图像处理算法。为了尽可能缩短部署时间并将注意力放在增值活动上，图像处理应用中常用的一种框架是 OpenCV。OpenCV 提供能够快速而轻松地部署的 C、C++ 和 Python 图像处理功能。这种方法可在部署流程中节省时间，因为它可使开发人员将注意力放在增值活动和专业算法上。

应对挑战

对接、性能、低延迟和决断力的挑战可利用各种单片系统解决，例如 Xilinx 汽车 XA Zynq UltraScale+ MPSoC。这些器件可提供各种处理系统和可编程逻辑，并结合任意互连功能以及功能加速的专用资源，例如机器学习推理。

将处理系统与编程逻辑彼此划分的方式让开发人员能够在可编程逻辑中部署复杂的图像处理管道，同时在 ARM CORTEX-A53 或 CORTEX-A72 之类的高性能处理器中运行高等级算法。

得益于可编程逻辑 I/O 的灵活性，Xilinx Zynq UltraScale+ 器件的 I/O Bank 可直接支持 MIPI DPHY — 不再需要外部 PHY，使用多个 MIPI 接口可节省板上面积并提供更紧凑的解决方案。可编程逻辑中的大量 I/O 引脚能够部署数量众多的 MIPI 接口，将设计人员从旧式的 CPU 和 GPU 解决方案的限制中释放出来。

在可编程逻辑设计中，MIPI 视频流可使用 MIPI-CSI2 IP 核心进行解码。MIPI-CSI2 IP 块的输出将处理为 AXI 流格式的视频，在可编程逻辑内使用标准流化接口，利用 Xilinx IP 库中的 IP 块、第三方 IP 提供商，或者在需要自定义算法时利用高层次综合，轻松地构建图像处理管道。这便能在可编程逻辑内创建真正的图像处理管道，显著减少延迟并提高决断力。

xfOpenCV 库

为了能够充分利用框架（如 OpenCV）中创建的高层级算法模型，Xilinx 提供了 xfOpenCV 库。xfOpenCV 库包含若干常用 OpenCV 函数，可以使用 Xilinx 统一软件开发工具 Vitis 中的高层次综合，将这些函数综合到可编程逻辑中。这样就可以运用 OpenCV 实现高等级建模，然后在可编程逻辑管道内快速而轻松地部署相同的函数，而无需编写硬件描述语言。

如果图像处理或下游处理需要涉及 H.264/H.265 编/解码，可以使用 XA Zynq UltraScale+ MPSoC 的 EV 版本，它内置视频编解码器单元，并提供了用于视频流缓冲的相关 UltraRAM。

很多自动驾驶车辆功能要求系统智能必须能够对图像处理算法管道检测到的物体进行识别和分类。对于 Xilinx 器件，Vitis AI 能够加速常用的 ML/AI 框架，包括 Caffe 和 TensorFlow。Vitis AI 还提供了 Model Zoo、AI Compiler、Optimizer、Quantizer 和 Profiler，将应用程序部署到深度学习处理单元上。

利用受支持的任何一种标准，如汽车以太网、CAN 或 CAN-FD，就能将从图像处理管道、机器学习或更高等级算法提取的信息，通过处理系统或可编程逻辑中可用的接口，传输至中央车辆控制器。

如果视觉系统的原型制造和测试需要高性能测试或显示接口，可以利用处理系统或可编程逻辑上可用的高速接口轻松地部署。

此类接口的示例包括 GigE VISION、10 Gig ETHERNET 和 PCIe，而 DISPLAYPORT 接口可用于从可编程逻辑输出实时视频或从处理系统输出经过处理的画面。

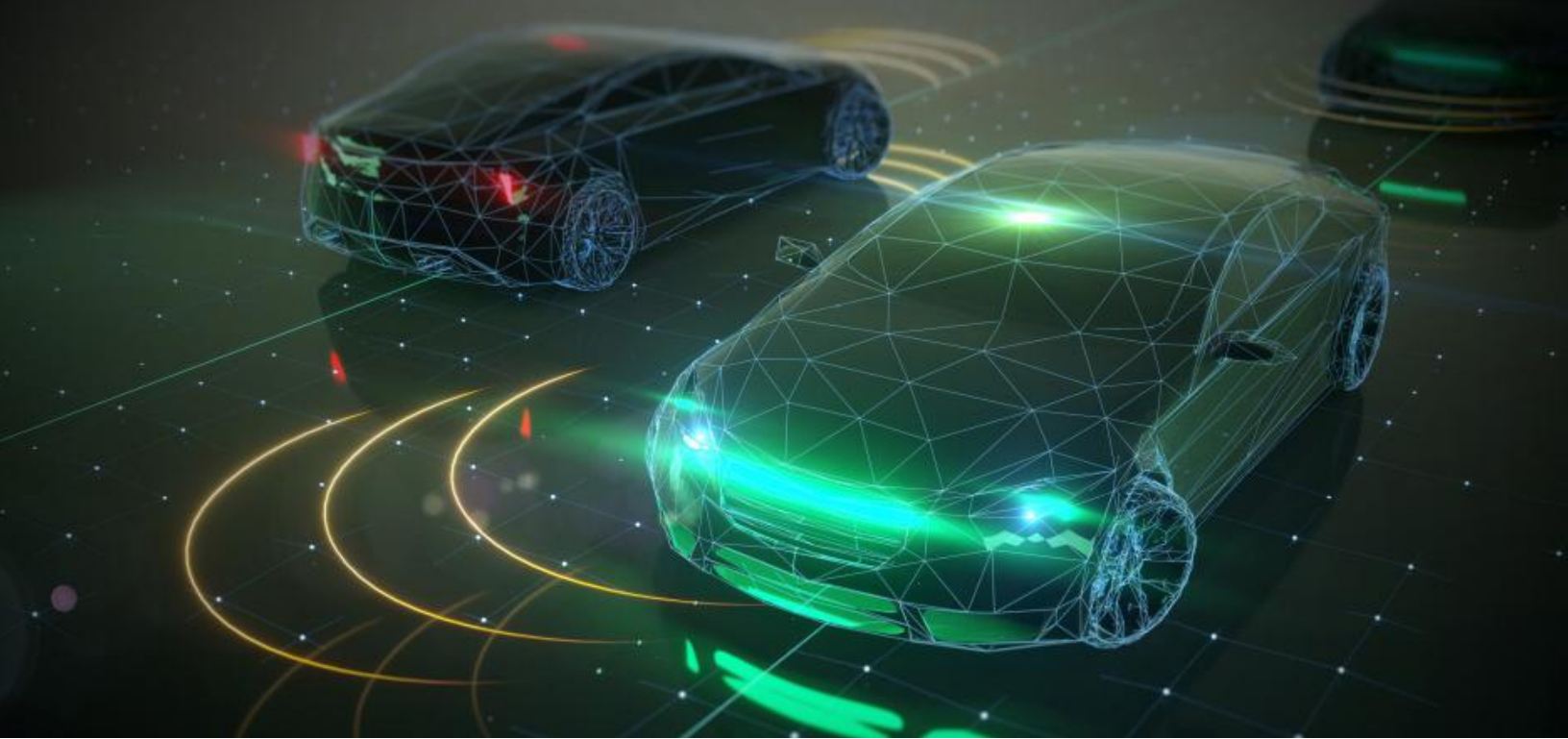
由于安全性是操作自动驾驶车辆、尤其是汽车视觉功能的核心，因此平台管理单元可在 XA Zynq UltraScale+ MPSoC 中用于监测器件的内部供电电压和核心温度。该器件还可在处理系统和可编程逻辑之间形成内部隔离以消除共同原因故障，使处理系统和可编程逻辑能够独立地相互监测性能和状态。

同样位于处理系统内的实时处理单元（RPU），设计时充分考虑了 ISO 26262 的应用，包含锁步运行的双 Arm Cortex-R5 处理器。如果需要，通过 RPU 能够直接在视觉处理系统内实施安全处理。典型示例可能包括监测摄像头和通信链的状态。为了支持应用处理器与实时处理器之间的通信，可以利用 OpenAMP 框架管理多处理器环境中的通信。

处理系统内提供的 SATA 和 PCIe 接口具有记录系统信息、甚至在必要时记录视频画面的能力，用于系统监测、日志记录和维护。

总结

视觉系统在自动驾驶车辆操作中扮演着关键角色，由于需要部署大量摄像头，它们本身受到多种限制。这些挑战包括对接、低延迟处理和决断力。通过各种 SoC，例如 XA Zynq UltraScale+ MPSoC，设计人员不仅能够利用器件的功能，还可以支持生态系统，以创建能够应对汽车视觉应用要求的对接、低延迟、更高决断力和安全性挑战的解决方案。



ADAS 雷达

简介

雷达是自动驾驶车辆操作中的技术基石。与激光雷达和视觉系统不同，雷达是唯一不受光线和天气条件（包括浓雾和大雨）影响的传感器技术。但是，为了提供足够准确的信息以实现更高 SAE 自主级别的操作，通常使用四维雷达。4D 雷达提供的信息不仅涉及方位角、高度和倾斜距离，还涉及多普勒频率。多普勒频率可用于确定要检测的目标速度。

因此，4D 雷达是一套可兼具大视野（100 度）、高空间分辨率（1 度）和较远探测距离（大约 300 m）的系统。辅以交通堵塞辅助、高速公路驾驶和自动泊车等功能，这有助于实现 SAE 2 级和 3 级的自动驾驶车辆。

架构功能和考虑因素

雷达系统需要生成、传送、接收和处理射频信号的能力。传统上，这需要使用基带数字处理来生成和处理信号，还需要使用射频前端以从基带进行升频和降频转换。使用模数转换器（ADC）和数模转换器（DAC）将信号转换为模拟量，以及将模拟量转换为信号。数字基带内的信号处理包括雷达线性调频的生成以及结果后处理。

但是，使用 4D 雷达时，经常通过单脉冲波束形成法在扫描中引导波束。每次扫描，在定义了时间长度的期限内连续地发出多个线性调频

雷达系统需要生成、传送、接收和处理射频信号的能力。

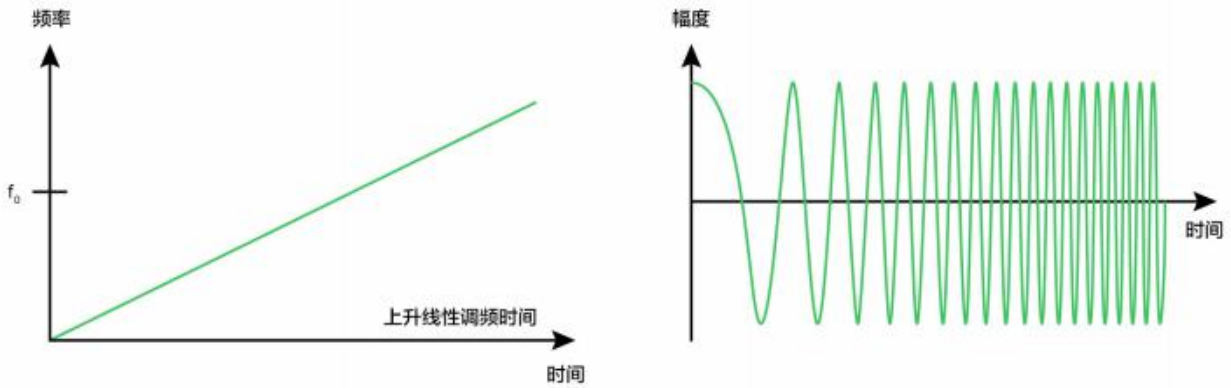


图 1: 线性调频信号

然后，基带处理能够利用二维 FFT 处理接收的线性调频。线性调频响应的第一个 FFT 提供范围信息 — 通常此 FFT 是两个中的较大者，例如 4096 点。来自第一个 FFT 的结果传递到第二个 FFT，它扫描完整时段并提供速度信息。第二个 FFT 比第一个要小，通常少于 1024 点。方位角和高度可从单脉冲接收器提供的总和及相差频道确定。

除了线性调频生成和 2D FFT，基带处理还需要能够执行降低和增加信号采样率、恒虚警率（CFAR）适应之类的算法以及与车辆中的下游模块的处理和通信，才能使车辆安全地应对识别的目标。

车辆在其环境中安全工作需要来自雷达的信息，因此在基带处理中应用的算法必须能够提供低延迟回应。

在原型设计和调试中或者在车辆信息娱乐系统中，也可能需要生成雷达系统可视场景。该场景应显示所检测目标的位置、速度和轨迹。

在电子层面，基带处理必须能够对接高速 ADC 和 DAC 器件。这会在电路板设计中带来风险，包括从低噪声模拟和射频部分高速路由和分割数字信号的完整性 — 实现这些的同时还必须确保整个系统紧凑的尺寸、重量、功率和成本（SWaP-C）目标。

满足这些性能和 SWaP-C 要求可能很有挑战性，需要在整体解决方案架构中做出牺牲。

应对挑战

Xilinx Zynq UltraScale+ RFSoc* 可以帮助设计人员应对性能和 SWaP-C 挑战。RFSoc 系列器件不仅包含采用高性能四核 Arm Cortex-A53 64 位处理器的处理系统，其可编程逻辑中还包括每秒千兆级采样的 ADC 和 DAC 以及必需的支持基础设施，包括升频和降频转换器、复杂混合器以及插值器和抽取器。这就极大地降低了与射频前端互连的复杂度，同时还降低了所需的总部署面积。

使用此类系统后，处理系统内的应用处理单元（APU）可部署高等级应用和通信框架，使下游车辆系统能够使用标准接口，例如 CAN 和汽车以太网。在 APU 内运行的应用程序可使用嵌入式 Linux 操作系统和 Xilinx PetaLinux 构建流。或者，也可通过适当的堆栈（例如，RTEMS 或 QNX）部署其他操作系统。

*Zynq UltraScale+ RFSoc 器件不是 Xilinx 汽车级产品组合的一部分。

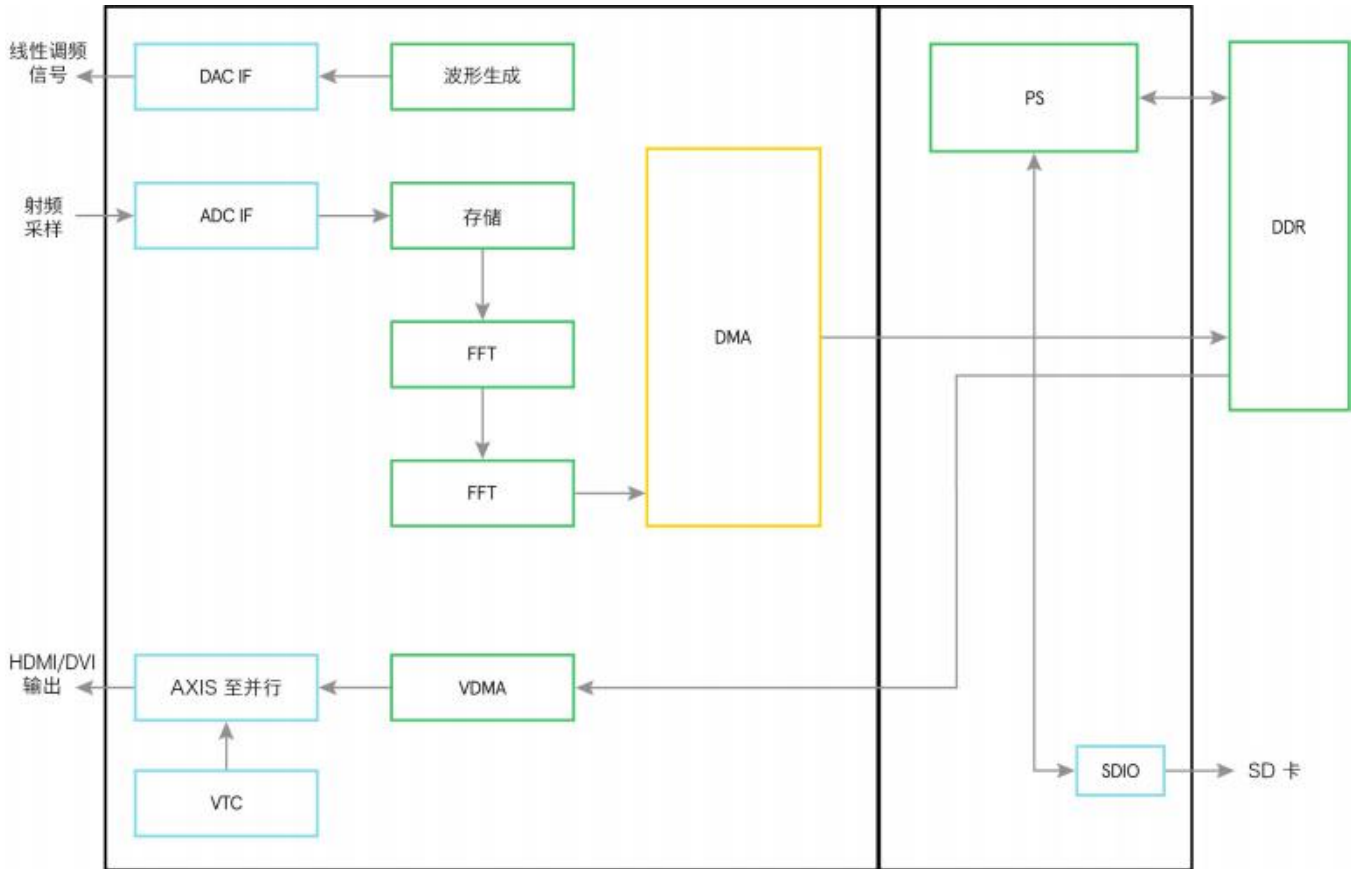


图 2 和 3: 如何通过 RFSoc 应对性能和 SWaP-C 挑战的框图

为了能够实时控制雷达系统，可以使用处理系统内的实时处理单元。RPU 旨在部署通过 ISO 26262 认证所需的安全关键功能，包含锁步双核 Arm Cortex-R5 处理器。RPU 内的这些处理器可用于执行安全功能，与系统监视器和平台管理单元一起监测单点和潜在故障。

支持低延迟解决方案获得的益处不仅包括可编程逻辑内的可配置逻辑块，还包括专用 DSP 元件以及从分布式 RAM 到 Block RAM 和 UltraRAM 的多种内存存储布局。

RFSoc 系列器件提供 3145 到 4272 之间的专用 48 位 DSP 元件，分布在可编程逻辑区域内。这些 DSP 元件设计为支持部署 FFT、收缩和多速率 FIR 滤波器、CIC 滤波器以及实时和复合倍增器和累加器。如果不需要 48 位分辨率，设计人员可以利用单指令多数据（SIMD）操作模式并执行双 24 位操作或四重 12 位操作。

可编程逻辑

通过器件的可编程逻辑元件可以实现线性调频和产生信号处理管道，包括 2D-FFT、CFAR 和位置提取。算法可借助编程逻辑能够平行实施，这种实施方式可响应低延迟和更高的确定性。

本地数据存储可获得更高的系统性能。 为了支持此功能，可以在可编程逻辑内使用 Block RAM 和 UltraRAM 元件。

Block RAM 非常适合用作算法阶段之间存储数据的小型缓冲区、存储器和 FIFO，而 UltraRAM 设计用于在需要存储更多的数据时替代外部存储器。UltraRAM 块比 4K x 72 位的 Block RAM 更大，每个块可提供 288Kb 的存储容量，而 Block RAM 仅为 36Kb。从架构上来说，UltraRAM 块可以级联后，构成大内部存储器，用于存储从 ADC 捕获的数据或 DAC 上的输出。在 RFSoc 系列器件内，UltraRAM 的容量范围在 13.5 Mb 到 22.5 Mb 之间，取决于器件。

RFSOC 的主要元件是 ADC 和 DAC 转换器，它们工作时的采样率可达到每秒千兆采样（GSPS）。

根据器件的代数不同，此采样率可在 2 GSPS（对于一代 ADC）至 5 GSPS（对于三代器件）之间变化，在辅助射频输入带宽下，能够跨全部四个 Nyquist（奈奎斯特）图区工作。一代 DAC 采样率为 6.5 GSPS，二代 DAC 采样率为 10 GSPS。这样的采样率，加之宽射频输入带宽，有助于显著降低射频收发路径的复杂度。

为了简化可编程逻辑内的基带处理，每种 ADC 或 DAC 都可获得数字升频和降频转换器、插值器和抽取器、复杂混合器和数字控制振荡器支持。这样可采用更高等级的系统方法配置射频前端，而非借助低等级的硬件描述语言（HDL）。

为了能在运行时进行系统优化和配置，ADC、DAC 和支持升频和降频转换器、混合器和 NCO 的设置可通过在处理系统上执行的软件完成

可编程逻辑内容的开发可利用随 Vivado 设计套件提供的庞大 IP 目录。或者，如果需要专业算法，可以使用 C、C++ 甚至 MATLAB 之类高等级语言实施。这些利用高等级合成工具的功能，例如 Vitis 统一软件平台。

如果系统需要可视雷达场景，可利用处理系统内的 DisplayPort 接口来产生此输出。DisplayPort 输出能够从处理器系统和可编程逻辑输出视频帧。或者，也可使用灵活的可编程逻辑 I/O 实施 HDMI/ DVI 或其他定制视频输出标准。

总结

通过处理器、可编程逻辑和射频数据转换器的紧密集成，打造的系统可获得汽车 4D 雷达所需的高性能和低延迟特点。借助支持设计工具生态系统，能够轻松地创建处理器、可编程逻辑和系统应用程序。

VITIS

Vitis 是一种统一软件开发环境，通过它能够创建在应用处理器单元和实时处理器单元（RPU）中运行的系统应用程序。Vitis 还支持 OpenCL，后者可用于在可编程逻辑中利用高等级合成创建加速内核。



汽车应用中的聚合、融合和加速

简介

自动驾驶是汽车开发领域最热门的主题之一，现有和新入行的汽车制造商都在开发自动驾驶车辆。这些开发人员可从摩尔定律受益，它在处理能力和传感器技术方面带来了明显提高，同时还降低了成本。

处理和传感器能力的这种提高非常重要，因为它能让车辆利用包括 GPS、雷达、激光雷达和视觉系统在内的各种传感器实时了解其所处环境并安全地与之交互。不同的传感器、算法和处理能力可实现不同等级的自主能力。

自动驾驶是一个复杂的领域，涵盖从全自动操控到与驾驶员共同控制等的宽广范围。汽车工程师协会（SAE）已经定义了自主能力的几个等级：

SAE 等级	名称	例子	车辆控制	监控	故障回退控制	车辆能力
0	无自动化	不适用	真人驾驶员	真人驾驶员	真人驾驶员	不适用
1	驾驶辅助	自适应巡航控制/车道保持和泊车辅助	真人驾驶员和车辆	真人驾驶员	真人驾驶员	部分驾驶模式
2	部分自动化	交通堵塞辅助	车辆	真人驾驶员	真人驾驶员	部分驾驶模式
3	有条件自动化	完全怠速熄火公路驾驶自动泊车	车辆	车辆	真人驾驶员	部分驾驶模式
4	高度自动化	自动驾驶	车辆	车辆	车辆	部分驾驶模式
5	全自动	无人驾驶	车辆	车辆	车辆	所有驾驶模式

图 1. SAE 自动驾驶能力的SAE等级

这些模式都能提供关于车辆整体环境的数据，而形成全貌需要将这些不同的元件融合在一起。不同的传感器部署和模式将会根据实施的自动驾驶等级而变化。但是，摄像头将用于车道偏离警告、盲点检测和交通信号识别之类的应用，而 4D 雷达可确定所跟踪物体的距离、速度和方向。

对于部分自动化（等级二）和更高等级，完全了解车辆所处环境的能力对于安全导航至关重要，因为车辆需要识别其位置和周围的障碍物。系统利用摄像头、雷达和激光雷达以及 GPS 数据了解其所处环境。仅凭 GPS 数据还不够，因为它的准确度会有变化，很容易被建筑物和基础设施阻挡。

使用不同的传感器模式对开发自动驾驶的人员提出了重大挑战。每种传感器模式都采用不同的格式并利用不同的接口（例如，MIPI、LVDS、SPI 或 I2C）提供其数据。

高带宽接口，例如雷达/激光雷达和视觉系统，需要进行重要的预处理以识别目标并对物体进行分类和检测。同时，这些高带宽接口需要专用的处理复合链，可能会影响整个系统的响应能力和确定性。

当处理并整理了来自不同传感器模式的所有信息后，更高等级的决策算法可对其进行处理，并利用它安全地与车辆环境交互。

如果失败，自动驾驶可能会导致人员死亡或受伤，并产生重大的环境破坏。为了尽可能地降低此类失败风险，系统的开发必须符合监管标准，例如 ISO 26262。

架构功能和考虑因素

部署自主能力的核心是中央处理系统。为了成功地部署自动驾驶，中央处理系统必须执行下列功能：

- **传感器接口和处理** — 处理单元与处理单元内的加速器之间的信息基本处理、路由和切换。后处理的示例包括图像预处理管道、物体检测和 4D 雷达后处理（例如，二维 FFT、CFAR 和轨迹提取）。
- **高性能串行处理** — 数据提取、传感器融合和高等级决策。在有些应用中，将在高性能串行处理内部署神经网络。
- **安全处理** — 根据分布式和并行数据库（DAPD）器件中的预处理功能检测到的环境和来自神经网络加速和高性能串行处理元件的结果进行的实时处理和车辆控制。DAPD 接口结合不同的传感器模式，执行处理单元与处理单元内的加速器之间的信息基本处理、路由和切换。
- **神经网络加速** — 加快进行物体检测和分类的神经网络的速度（例如，行人和车辆检测，以及路标分类）。

根据应用需要，这些功能可以跨多个器件执行，器件利用汽车以太网之类的汽车通信标准连接。

挑战

随着自动驾驶系统量产，它们必须能够达到制造商确定的尺寸、重量、功率和成本（SWaP-C）目标。自动驾驶开发人员面临的这些挑战可以归纳为：

- 与多种高带宽传感器模式对接的功能
- 对高带宽传感器数据（例如，图像后处理管道、4D 雷达等）执行复杂后处理算法的功能
- 实时执行复杂后处理和高等级决策的处理能力
- 满足 ISO 26262 之类标准的能力
- 达到苛刻的 SWaP-C 目标

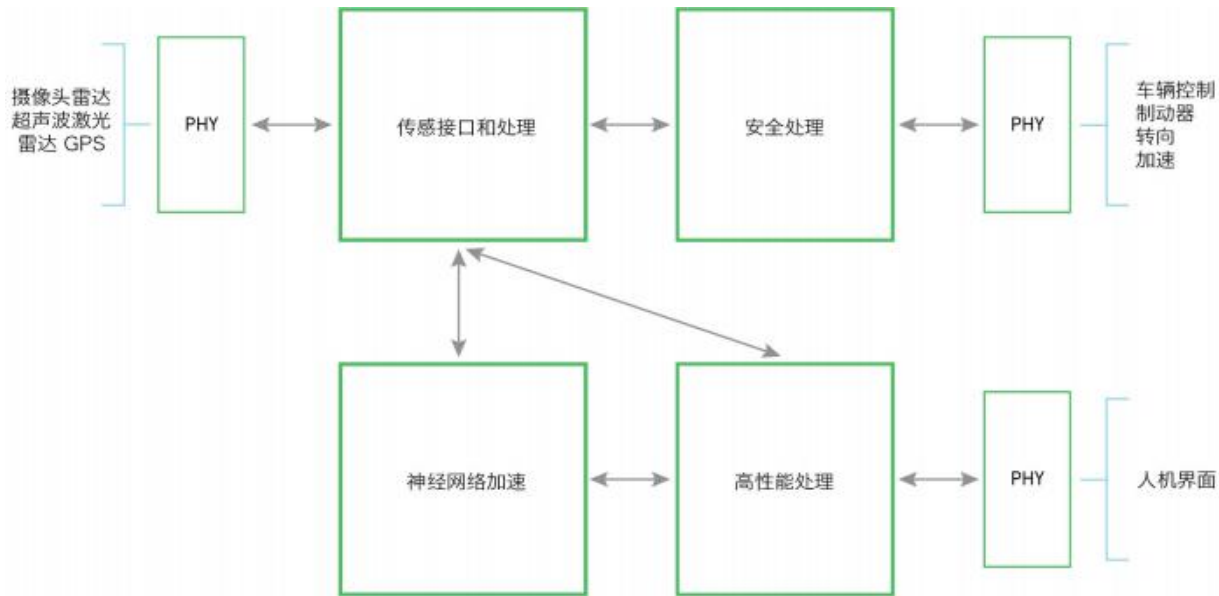


图 2.使用汽车以太网连接建立连接的器件

应对挑战

XA Zynq UltraScale+ MPSoCs 提供的不仅是传感器接口和处理，还在同一晶片内提供高性能、安全处理和神经网络加速，并可随着 SAE 等级升高进行扩展以支持多器件实施。这种高度集成和可扩展解决方案显著地改善了 SWaP-C。

对于实时控制，Zynq UltraScale+ MPSoC 还提供：

1. 实时处理单元（RPU），它包含锁步双 Arm Cortex-R5 处理器，能够执行高达 ASIL C 标准的安全功能，适合对安全要求苛刻的应用。为了达到必需的功能安全性，RPU 设计为能够减少、检测和缓解单次随机失败，包括硬件和单事件诱发型。这些器件能够在处理器系统资源和/或可编程逻辑之间充分地分割功能。

传感器接口和处理带来的一个关键挑战是能够与各种高带宽传感器模态对接，它们都会采用不同的接口标准。

典型的解决方案将与采用 MIPI、JESD204B、LVDS 和 GigE（用于摄像头、雷达和激光雷达等高带宽接口）之类高速接口的各种传感器模态对接。传感器接口和处理还必须能够与较低速的接口对接，例如 CAN、SPI、I2C 和 UART。

2. 部署安全处理器带来了另一种挑战。安全处理器必须处理从 DAPD 和高性能串行处理接收的命令，才能安全进行车辆导航。安全处理器直接与车辆控制功能交互，例如转向、加速和刹车 — 这是自动驾驶的关键领域，因为这方面的错误可能会造成丧生和破坏环境。Xilinx 汽车级 Zynq UltraScale+ MPSoC 的 RPU 中包含双锁步 Arm Cortex-R5 核心，可用于执行安全处理功能。

除了 RPU 核心的锁步功能，Xilinx Zynq UltraScale+ MPSoC 中的其他缓解措施包括为 RPU 引入了纠错代码，与内存和缓存紧密契合，同时 DDR 内存通过双检错和单纠错代码进行保护。

3. 对各种工业标准接口的支持，包括 CAN、SPI、I2C、UART 和 GigE 等，同时 PL I/O 的灵活性使其能够直接连接 MIPI、LVDS 和千兆级串行链路，从而允许在 PL 内（通常使用 IP 核心）实现更高层级的协议。PL 内的协议实现还有助于轻松纳入标准的修订版本，还能灵活调整解决方案内支持的特定传感器接口的数量。PL 还让您能够在硬件设计中布置正确的物理层以便部署任何接口，从而拥有真正的任意互连能力。
4. 利用 Xilinx DNNDK 和 Vitis AI 加速神经网络。算法也可利用 Vitis 进行加速，这样可将 OpenCL 内核部署在可编程逻辑中。

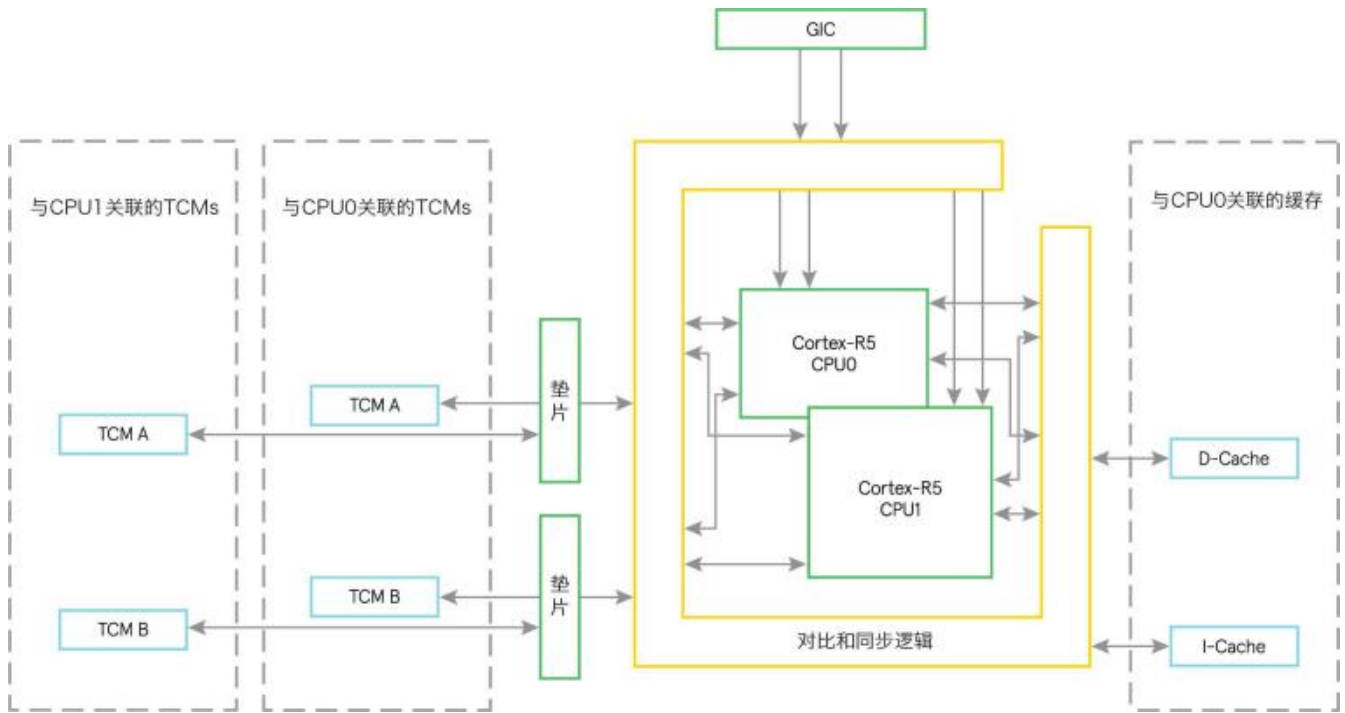


图 3. 实时处理单元架构

5. 开机内置自检（BIST）功能将确保工作之前基础硬件没有故障。在运行期间，还可根据用户的请求执行附加 BIST。Zynq UltraScale+ MPSoC 的架构还可能让您在器件内实现内存和外设功能隔离。
6. 重要的安全功能，包括：
 - 隐私安全配置（AES 加密）
 - 密钥滚动，用于防范差分功耗分析
 - 配置安全单元，用于管理器件的安全性状态

在 Xilinx 汽车级 Zynq UltraScale+ MPSoC 内融合这些特性之后，能够将安全处理与传感器接口和处理、高性能处理和神经网络加速部署在相同的晶片中。如果架构要求跨器件隔离，还可提供可扩展的解决方案。

当然，更高的集成度还可降低 PCB 设计的复杂性以及最终解决方案中所需的互连，同时提供功耗更低的解决方案。

总结

提供自动驾驶能力需要部署中央处理模块，它面临着多个挑战，包括与各种传感器模态对接以及传统的 SWaP-C 问题。利用 Xilinx 汽车级 Zynq UltraScale+ MPSoC 获得传感器对接、神经系统加速器和安全处理器功能的高度集成解决方案，能够产生外形更小、重量更轻、能效更高的解决方案。



汽车功能安全

简介

汽车系统制造商面临的一个严峻挑战是设计符合相关安全和质量标准的系统。

在汽车行业的功能安全中，这表示符合 ISO 26262 汽车安全完整性等级 (ASIL) 之一。作为基于风险的安全标准，ISO 26262 适用于车辆生产中的电气和/或电子系统，可能涉及驾驶辅助、推进力和车辆动态控制系统。四个 ASIL 表示不同的系统风险等级。

ASIL	关键性	失效时间	度量指标	后果
ASIL-D	最关键	< 10 FIT	单点故障指标 > 99% 潜在故障指标 > 90%	可能造成社区级人员死亡
ASIL-C		< 100 FIT	单点故障指标 > 97% 潜在故障指标 > 80%	可能造成重伤或致命
ASIL-B		< 100 FIT	单点故障指标 > 90% 潜在故障指标 > 60%	可能致命
ASIL-A	最不关键	< 1000 FIT	单点故障指标 < 90% 潜在故障指标 < 60%	可能造成轻微伤
QM	质量管理	不适用	非安全相关	

图 1. 汽车安全完整性等级

单点故障指标表示安全设计预防单点故障所致的风险的能力，而潜在故障指标涉及安全设计未发现的多点故障。

系统的 ASIL 等级通过执行风险分析确定，其中涉及：

- **严重性** — 对乘员造成的潜在人身伤害是什么？
- **暴露程度** — 系统/车辆暴露于风险的频率？
- **可控性** — 驾驶员/车辆可采取什么措施来预防人身伤害？

车辆中的每个系统将拥有不同的 ASIL 等级。例如，刹车、转向和安全气囊之类的关键系统通常为 ASIL-D 级，而大灯和 ADAS 系统通常为 ASIL-B 级。

打造达到相应 ASIL 等级的设计需要进行有效的设计分析和失效模式考量，以便检测并降低单点和潜在失效。

方法和挑战

达到所需的 ASIL 等级需要在架构设计层面考虑，从一开始就包含必要的缓解和监测元件。在开发周期中后来增加此类缓解和保护措施可能会显著地影响项目的进程和成本。

设计的重要元素之一是从设计中尽可能更多地消除单点故障。在架构层面，这可能意味着在设计中实施硬件容错或多样化的电路元件。

电子系统中的典型单点故障包括输入和输出、时钟、电源、重置和电源定序以及配置内存。如果故障能够传播并影响硬件容错或多种通路，这些领域的故障可能会导致整个系统发生故障。

单点故障	缓解措施
输入/输出	重复
时钟	多个时钟
电源	独立电源
重置和定序分隔	重置和定序系统
配置内存	检错和纠错
处理器	锁步运行
逻辑设计	三重模块冗余

图 2. 单点故障和缓解措施

设计人员还需要考虑故障在设计内传播造成的影响 — 利用物理隔离之类的技术防止单个故障影响缓解电路元件。

达到所需的 ASIL 等级需要考虑的不仅仅是必须分析和验证的电路设计。同样重要的是为处理器、FPGA 或其他可编程器件创建解决方案所用的工具，必要时应考虑对其进行认证。此类考量非常关键，因为工具中的错误也可能进入设计或导致无法检测到设计中的错误。

在 ISO 26262 领域，是否需要认证电子设计自动化（EDA）工具在于其工具可信等级指示。为了确定工具可信等级，必须对所用的每种 EDA 工具分析下列参数：

- 出现故障的软件工具及其错误输出，是否会导致指定的待开发安全相关软件，违反某些安全要求
- 预防或检测工具输出中的此类错误的可能性

为了帮助确定工具可信等级，我们可以使用工具影响（TI）和工具检错（TD）。使用 TI 等级一，除非错误不可能由该工具产生，在这种情况下应使用 TI 等级零。

工具检错	可信度
TD4	否
TD3	低
TD2	中
TD1	高

图 3. 工具检错和可信度

TI 和 TD 用于确定工具可信等级，它指示工具是否需要认证。

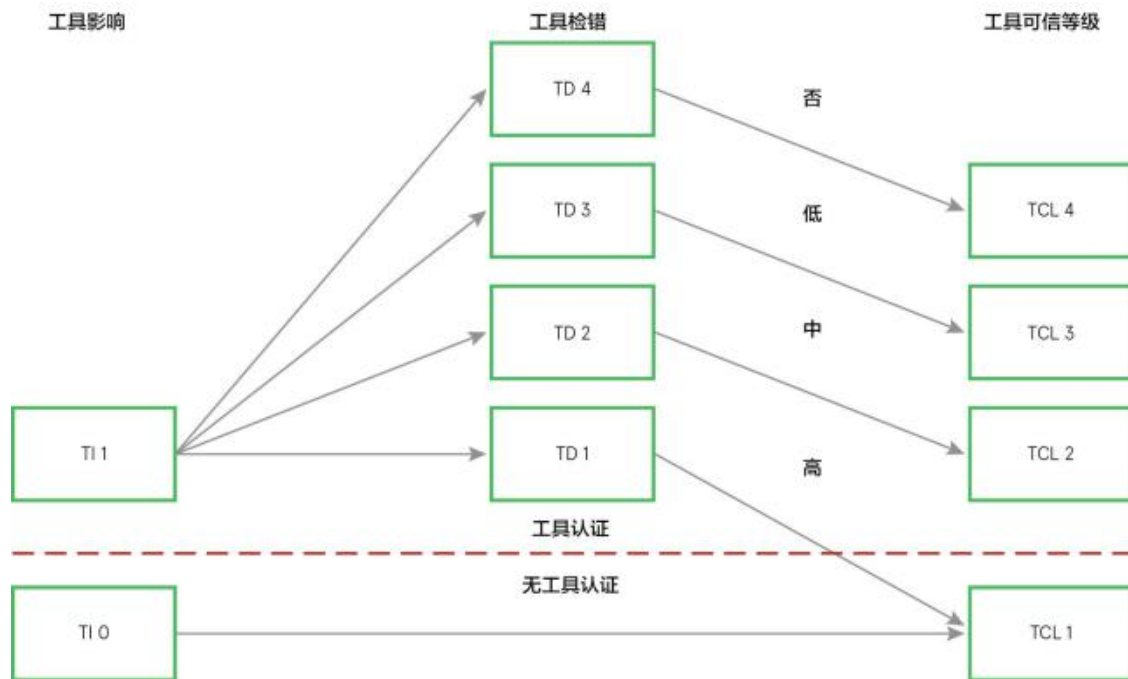


图 4. 确定工具可信等级

当然，当系统中使用 FPGA 或 SoC 时，ASIL 解决方案的器件选择、架构设计、设计实施和工具链选择都会变得具有挑战性。

由于其高性能、低延迟、更高的确定性和能效，FPGA 和 SoC 广泛地用于多种汽车系统，包括雷达、视觉类系统、车辆通信和 ADAS（自动驾驶辅助系统）。

应对挑战

必须满足 ASIL 要求的汽车系统的设计人员面临的许多挑战可通过 Xilinx 汽车 XA Artix-7 FPGA、XA Spartan-7 FPGA、XA Zynq-7000 SoC 和 XA Zynq UltraScale+ MPSoC 系统器件以及相关工具链得到解决。

合适器件的选择取决于最终应用、对接、处理能力和资源要求。但是，XA 产品组合中的所有器件都根据 AEC-Q100 进行了部件认证测试。提升部件品质有助于确保全面的可靠性，进而有助于提升系统的功能安全性。

XILINX 提供了多种知识产权核心、器件架构和文档支持以帮助达到所需的功能安全性等级。

为确保可靠的逻辑设计，在可编程逻辑内提供了低层级支持，包括 Block RAM 上的检错和纠错。三重模块化冗余（TMR）或锁步 MicroBlaze 软核处理器经常部署在可编程逻辑中以提供多样化的实现。XADC 和 Sysmon 则可用于监控设备供电线路和核心温度，并在温度超限时告警。

可编程逻辑内的功能之间的隔离可利用 Xilinx 隔离设计流和 Vivado 隔离验证器实现。

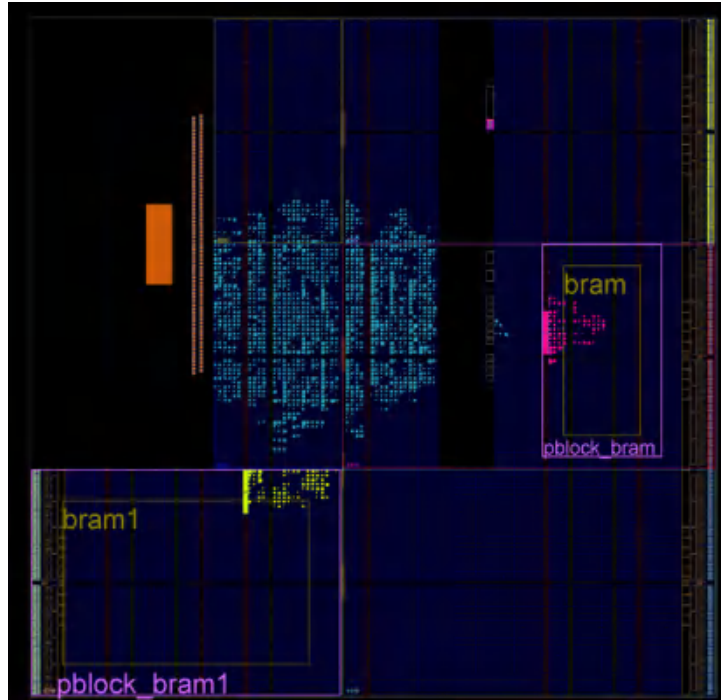


图 5. BRAM 和 BRAM 控制器的隔离

为了保护可编程逻辑配置，提供了必要位技术，用于报告配置流内的必要配置位的数量。该技术可与单事件缓解 IP 核心共用以监测和纠正器件配置内存中的错误。

XA Zynq UltraScale+ MPSoC 器械架构和安全手册通过了 ISO 26262:2011 认证，为 ASIL-C 级。能够实施 ASIL-C 解决方案的架构特点包括：

- 低功耗、全功耗和可编程逻辑中的三个独立域，各有自己的独立电源和时钟确保硬件容错
- 平台管理单元中的 TMR 启动安全、电源和错误管理处理器
- 低功耗域中的锁步 Arm Cortex-R5 处理器
- 内存和外设保护单元能够在处理系统内实现隔离
- 关键内存上的纠错代码
- 可测试的架构，包括逻辑和存储器内置自检、错误注入和软件测试库。

驻留于 XA Zynq UltraScale+ MPSoC 中的使用硬件容错的 ASIL-C 应用实施示例之一：实施两个安全通道，一个位于低功耗域内，另一个位于可编程逻辑中。每个安全通道都将对接到传感器，处理结果可进行比较和投票。通过此类方法可实现图 6 中所示的多样化容错硬件解决方案。

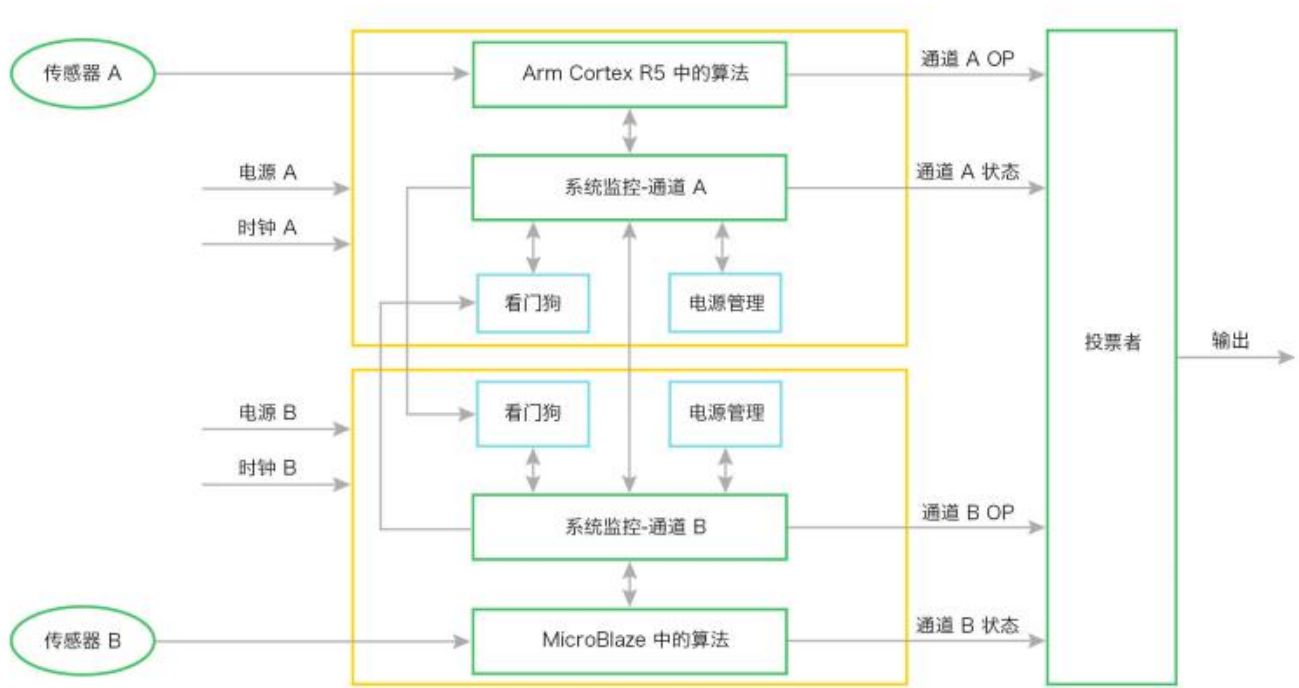


图 6. 多样化的容错硬件解决方案

支持 Xilinx 汽车器件的工具通过了 ISO 26262-8:2011 认证。

经认证的工具让开发人员能够缩短总体开发时间 — 有助于确保项目的交付按时、高质量且成本效益出色。

有些 Xilinx 系统和工具旨在帮助达到认证要求，其中包括安全手册、软件安全用户指南、FMEDA 工具和实例，以及定期的可靠性报告。这些资源可通过 Xilinx Functional Safety Package 获得。

总结

为了达到 ISO 26262 规定的安全标准，需要考虑的不仅是整体系统架构，还包括实施技术的能力和鉴定。Xilinx 汽车器件按照 AEC-Q100 质量标准进行了鉴定，并且设计中充分考虑了 ISO 26262 认证，已在示例应用和认证工具链中展示。这样可让您自信地利用 Xilinx 汽车部件开发 ISO 26262 解决方案。



乘员监测

简介

提供安全的汽车体验要求车辆不仅监测和观察其外部环境，还得能够监测其车厢内部空间，并且考虑到延迟。此类监测可以提高安全性 — 例如，通过监测驾驶员警觉性或通过其他方式调节车内环境以使乘员感到舒适。

处理视频和音频指令并实施控制算法和反馈机制需要消耗可观的处理能力，才能使这些功能接近实时地执行。在手势识别中，反应缓慢的交互将会无法令人满意和沮丧，如果系统未与疲惫的驾驶员交互，甚至可能引发潜在的安全事件。

架构功能和考虑因素

内部监测系统执行的最关键的乘员监测功能是监测驾驶员的警觉性和睡意。

像这样的监测算法要求图像处理管道连接到摄像头并接收图像。来自摄像头的图像需经进一步处理以检测驾驶员的面部和眼睛，然后再实施眨眼检测。

典型的处理阶段将包括从 RGB 到灰阶的色度空间变换，以及实施面部、眼睛和眨眼检测之前的直方图均衡化。眨眼检测必须进行定制以适应不同的响应时间（典型的人类眨眼响应为 400 ms）。

卷积神经网络带来检测面部和眼睛的功能。

利用视野更宽广的摄像头监测车内的其他乘员及其行为。进一步处理该视频以支持对乘员的手势检测和识别。通过视频帧检测这些手势需要使用卷积神经网络。

内部监测

实施内部监测需要从专注于驾驶员的摄像头到能够观察整个车厢及车内所有人员的广角摄像头在内的各种传感器。车厢设置可通过检测乘员或乘员的手势或命令确定。还可能需要其他传感器，例如调节舒适度的温度传感器、接收语音指令的定向麦克风、接触式传感器和控制器上的触觉反馈装置。

系统需要能够对接到并处理多个定向麦克风。

乘员监测系统还可利用语音指令控制车厢内的系统。用户可利用高级系统请求查看信息和状态，例如到达目的地的时间和当前交通状况。虽然基本语音处理可在解决方案内实施，但是交通感知之类的高级功能需要连接到云。

解读手势或语音指令后，系统必须能够对接控制器以提醒昏昏欲睡的驾驶员，或根据个人识别结果调节温度设置或座椅位置。这要求能够驱动马达、传感器和触觉反馈装置，全都接近实时响应。

车厢状态及其控制器和乘员也可能会受到高分辨率触摸屏影响。这样能够实现状态报告并配置手势控制，以及所有控制器备份设置的功能。

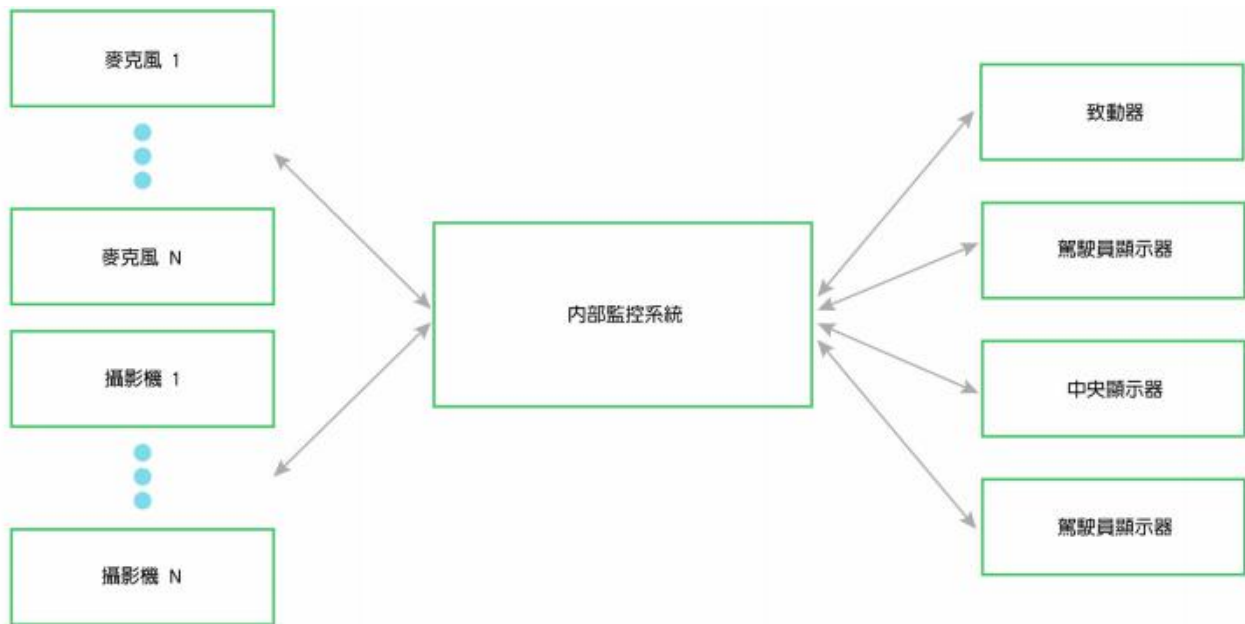


图 1. 乘员监控系统

应对挑战

Xilinx XA UltraScale+ MPSoC 的各种单片系统有助于以更低功耗和成本的解决方案获得所需的架构。

紧密契合的处理系统和可编程逻辑域可用于部署乘员监测系统，从而实现低延迟和高能效解决方案。

可编程逻辑域将实施摄像头对接和图像处理管道。得益于可编程逻辑的灵活性，可以轻松地支持多种摄像头类型，包括 MIPI、CameraLink、VGA 及自定义类型。

利用接口

设计人员可以利用处理器系统和可编程逻辑提供的对接功能。处理系统可以对接多种使用公共嵌入式接口（例如，I2C、SPI 和 UART）的传感器，而可编程逻辑可用于对接使用 I2S 接口的定向麦克风。

这些图像处理管道可并行部署，从而减少总体处理延迟。当图像处理管道完成时，结果传输到处理器域以进入更高等级的神经网络应用。在可编程逻辑内创建图像处理管道时可利用 Vivado 设计套件库中的众多 Xilinx 和第三方 IP 核心。如果需要定制 IP，设计人员可使用硬件描述语言、高等级合成工具或 Xilinx System Generator 或 Model Composer 之类利用 Simulink 的高等级开发工具创建。为了能够在框架（如 OpenCV）中创建高层级算法模型，Xilinx 提供了 xfOpenCV 库。xfOpenCV 库包含多种常用 OpenCV 函数，可以使用 Xilinx 统一软件开发平台 Vitis 中的高层次综合功能将其综合到可编程逻辑中。这样能够使用 OpenCV 进行高等级建模 — 然后在可编程逻辑管道内快速而轻松地部署相同的函数，而无需编写硬件描述语言代码。

神经网络的实现可以借助 Vitis AI。Vitis AI 能够利用可编程逻辑加速常用的 ML/AI 框架，包括 Caffe 和 TensorFlow。

为此，Vitis AI 提供了 Model Zoo、AI Compiler、Optimizer、Quantizer 和 Profiler，用于将应用程序部署到深度学习处理单元，以便将面部和眼睛识别检测网络用于驾驶员检测。还可部署其他单元用于乘员手势识别功能。

马达和执行器控制经常需要高等级驱动电路，在 MPSoC 的控制下发挥作用。这样，我们可以在处理系统内使用三重计时器生成用于驱动马达 H 桥的脉冲宽度调制波形，以便控制转速的速度和方向。如果控制伺服马达时不使用 H 桥，可以实施相似的方法。

为了提供用户显示器，可以利用处理系统中可获得的 DisplayPort 输出实施低等级的视频输出。如果显示器接口要求 HDMI 或更低等级，例如 LVDS 或并行视频，此输出流可以在可编程逻辑内实施。在这两种情况下，视频可由处理器生成，也可从可编程逻辑实时馈送。

对于这样的应用，常见的是运行嵌入式 Linux 之类的高等级操作系统。为了能够创建适应可编程逻辑设计中所实施的设计的嵌入式 Linux 解决方案，Xilinx 在 PetaLinux 中提供了嵌入式 Linux 系统。PetaLinux 简化了 Yocto 搭建流，让开发人员能够自定义所需的应用程序、驱动程序和框架。

通过在 APU 核心上运行 PetaLinux，能够将 Vitis AI 与深度神经网络处理器单元（DPU）一起部署到可编程逻辑中。APU 还可用于创建用户显示器 — 可以利用 Qt 之类的框架生成专业外观的显示器。这种接口还将能够从触摸屏位置接收指令并显示高分辨率图形。

振动或触觉反馈所需的实时控制可通过处理系统中的 RPU 实现。APU 与 RPU 之间的通信可利用 OpenAMP 框架实现。



图 2. OpenAMP 框架

总结

通过监测车辆乘员，不仅能够更安全地进行车辆交互，还能营造更加舒适的乘员体验。利用 Xilinx 汽车 XA Zynq UltraScale+ MPSoC 器件以及能够使用 OpenCV、TensorFlow 和 Caffe 之类常用框架的支持开发和软件生态系统，可以实施更低能耗和成本的解决方案。



车联网（V2X）通信

简介

车辆相互之间、与其他道路用户之间以及与基础设施或路边设备（例如，交通标志、人行横道和红绿灯）之间进行通信的能力被视为提高道路安全性的柱石之一。

这些智能交通通信（ITC）系统均为短程。鉴于对安全的重要性，它们必须为低延迟通信，并且能够交流例如碰撞或红绿灯快速转变之类的事件。

目前，车辆中使用两种标准来达到此通信标准。第一种是专用短程通信（DSRC），它基于 Wi-Fi 并使用 IEEE 802.11 标准。另一种标准基于新兴的 5G 移动通信标准，使用 5G 的一个重要原因预计就在于其可靠的低延迟通信特点。

这两种通信标准各有其优缺点。DSRC 免费提供，代价是数据率较低，而 5G 解决方案需要费用，但是通信速度明显更快且延迟更低。

除了获取成本数据率之外，一个重要的不同之处是通信范围。DSRC 的范围为 300 米，而 5G 将可达到更大的范围。

由于两种解决方案都能接入车辆系统 — 影响其功能并最终影响车辆和乘员的安全性 — 因此解决方案的环境安全和网络安全至关重要。

系统制造

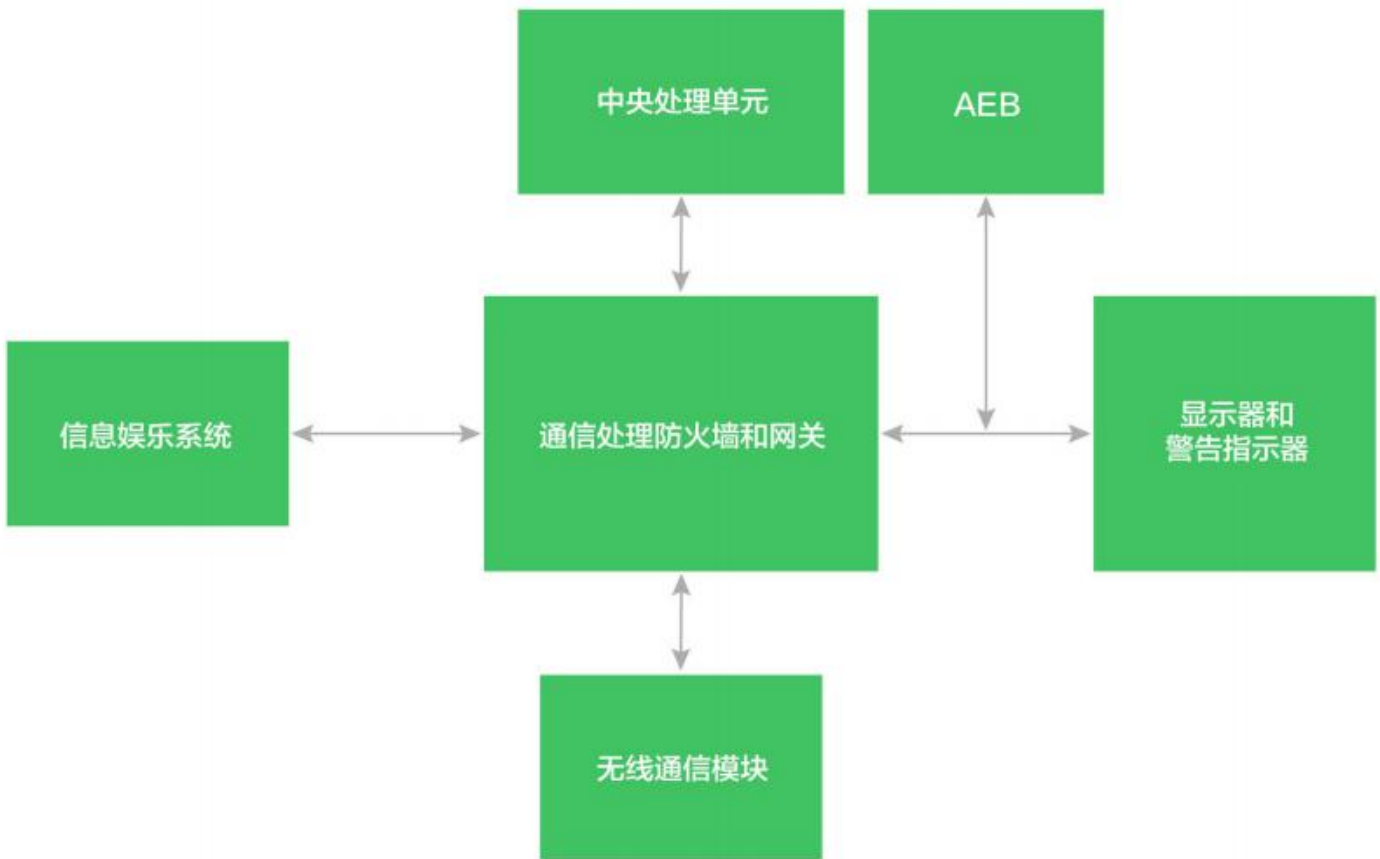
DSRC 和 5G 解决方案都带来了多种挑战，包括：

- 在 2020 年满足每辆车大约 \$350 USD 的单位成本（预计到 2058 年可降低到 \$200 USD）
- 能够适应不同地理位置的不同频率分配
- 达到汽车应用中要求苛刻的外型规格和功率预算
- 作为关键系统，不仅达到要求的性能，还必须达到可靠性目标
- 跨区域对适应能力进行标准化 - 如果部署在互补的使用案例中，它们可能需要同时支持 DSRC 和 5G 解决方案
- 适应新生产车辆和现有车辆的适用性

架构功能和考虑因素

典型 V2X 模块的架构将由若干不同的功能元件构成，包括：

- 通信处理模块（CPM），系统的中央处理枢纽，对接一个或多个射频前端以提供 DSRC 和 5G 通信堆栈。除了实施通信堆栈，CPM 还将连接到现有汽车车辆网络中，包括 CAN、CAN-FD、FlexRay 和汽车以太网。CPM 的关键角色之一是利用网关和防火墙实施物理和网络安全策略，以及身份验证和安全技术，以防止未经授权访问并确保数据安全。
- 无线通信模块（WCM），连接到 CPM 以实施通信标准的射频前端和下部元素
- 对接，这种功能是指通过现有汽车标准（例如，CAN、CAN-FD、汽车以太网、FlexRay）进行连接，以便与信息娱乐系统、头顶显示器和用于自主功能的中央处理单元进行通信。



XA Zynq UltraScale+ MPSoC 器件提供的功能可轻松地满足功能、安全和功率要求。

应对挑战

Zynq-7000 SoCs 和 Zynq UltraScale+ MPSoCs 可作为通信处理模块的中央枢纽。

这种处理系统和可编程逻辑的组合能够利用外部射频模块实现众多对接解决方案。处理系统支持 SDIO 之类的标准，它常用于 Wi-Fi 对接，而可编程逻辑可通过正确的外部 PHY 实现任意互联。这正是 5G 射频调制解调器/模块支持所需的。

通过处理系统和可编程逻辑还可利用各种接口标准连接现有的车辆网络，包括下表中所示的标准。该表概述了处理系统支持的主要接口，或者 IP 核心在可编程逻辑内的集成中的可用性。

这些接口使得 CPM 能够连接到信息娱乐系统、头顶显示器和车辆的中央处理单元。

接口标准	处理系统	可编程逻辑
CAN-FD		x
CAN	x	
局域互连网络 (使用 UART)	x	
千兆以太网	x	
汽车以太网		x
I2C	x	
SPI	x	
SDIO	x	
时间敏感以太网		x
DisplayPort	x	
HDMI 输出		x

表 1. XA Zynq UltraScale+ MPSoC 对接支持

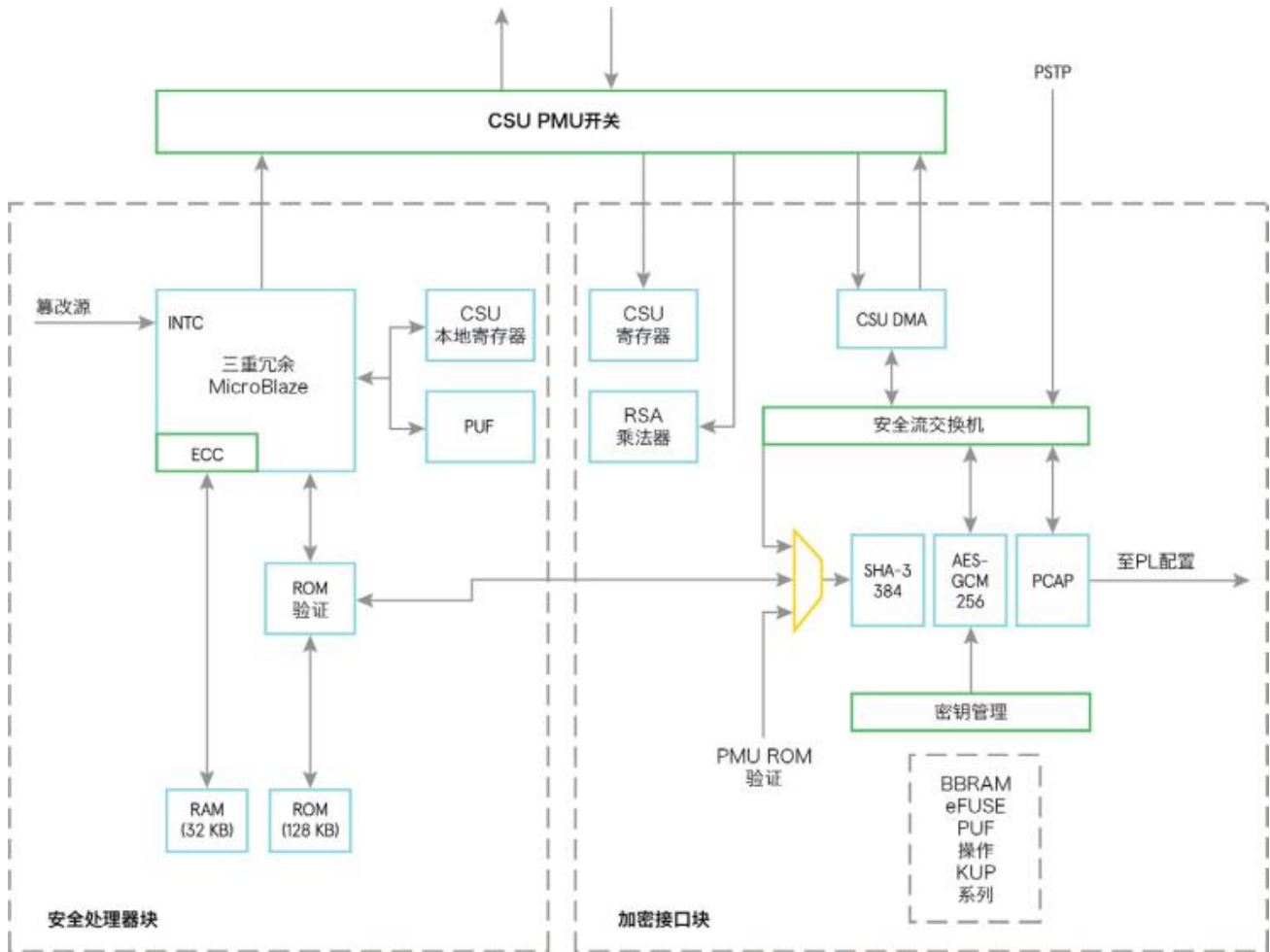
实施安全通信所需的算法、网关和防火墙在处理系统和可编程逻辑中均可实施。在可编程逻辑中实施不仅可带来灵活性，还能提高确定性，减少延迟和处理量。使用可编程逻辑还能使防火墙和网关的实施比传统的软件型方法更加安全。为了支持防火墙和网关的实施，处理系统提供了内存和外设保护功能，这得益于 XMPU 和 XPPU，利用它们可在创建并相互隔离外设和内存的情况下创建缝隙。

此外，还可利用 XILINX 隔离流在可编程逻辑内实施物理隔离。

CSU 在处理系统内创建分层安全实施，从而进一步巩固安全解决方案。这样可使器件借助 CSU 实现安全配置，后者支持 AES 256-GCM、4096 RSA Multiplier 以及为解决方案提供保密性、验证和完整性的 SHA-384。

支持运行时安全包括，通过内置系统监控器提供防篡改响应，这样就能够监控设备电压和核心温度，并在温度超限时发出报警。系统监控器还拥有外部连接，这些连接可用于增强外壳层的物理防篡改保护。

CSU 加密引擎的 AES、RSA 和 SHA 块还可以在运行时用来实现保密性、验证和完整性功能。由于 DMA 在 CSU 中可用，因此可利用其实现极其高效的处理。CSU 还能够实现密钥管理，包括提供执行密钥滚动以防御差分功耗分析攻击的密钥破解。



处理系统还能实施 Arm 信任区，在处理系统和可编程逻辑中产生相互垂直的安全世界和非安全世界。

为通信模块开发算法时，可以利用器件中的处理系统实施通信堆栈，同时还可利用可编程逻辑卸载加速功能以及防火墙和网关，包括为信息娱乐系统生成音频和视频流的功能。对于 XA Zynq UltraScale+ MPSoC 器件的 EV 版本尤其如此，它所包含的 H.264 / H.265 视频编解码器可用于对视频流进行编码和解码。

CPM 的整体软件解决方案可使用统一软件开发平台 Vitis 来实施。通过 Vitis，可将软件作为适用于 APU、RPU 和 PMU 的完备系统进行开发。

在此系统解决方案内，设计人员可以利用嵌入式操作系统（例如，APU 上的嵌入式 Linux）或实时操作系统（例如，FreeRTOS）创建整体软件解决方案。为了支持多处理器环境，可以利用 OpenAMP 在 APU 和 RPU 之间启用安全通信以及处理器间中断（IPI），并支持与每个 IPI 关联的短消息缓冲区。

除了 Vitis 提供的系统开发功能之外，Vitis 还可为使用 OpenCL 的加速提供支持。

总结

有些领域的标准仍不确定。变化的标准意味着解决方案需要能够进行适应。解决方案提供者可利用 SoC 结合可编程逻辑满足这种灵活性要求，同时提供功能、安全性、对接能力以应对 V2X 应用需要解决的 SWaP-C 挑战。

OpenCL 已启用

OpenCL 允许在 APU 的控制下将加速内核部署在可编程逻辑内。这意味着算法加速不一定非要由工程师的硬件描述语言（HDL）实施，还可使用基于 C / C++ 的高层次综合来编程算法。然后，可以使用杂注在算法中实施性能优化，以利用可编程逻辑的平行架构。

有关安富利的更多信息，请访问
WWW.AVNET.COM/APAC